AD-A058 865 COMMAND AND CONTROL TECHNICAL CENTER WASHINGTON D C F/G 9/2
NMCS INFORMATION PROCESSING SYSTEM 360 FORMATTED FILE SYSTEM (N--ETC(U)
SEP 78

UNCLASSIFIED CCTC-CSM-8D-15-78 NL

| OF |
AD
A058865

END
DATE
FILMED
11-78
DDC

C
C
T
C

(12) B.S.

COMPUTER SYSTEM MANUAL
CSM | GD 15-78
1 SEPTEMBER 1978

LEVEL II

COMMAND & CONTROL TECHNICAL CENTER

NMCS INFORMATION PROCESSING SYSTEM 360 FORMATTED FILE SYSTEM (NIPS 360 FFS)

GENERAL DESCRIPTION

78 08 22 035

DEFENSE COMMUNICATIONS AGENCY

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

COMMAND AND CONTROL TECHNICAL CENTER

Computer System Manual Number CSM GD 15-78

1 September 1978

NMCS INFORMATION PROCESSING SYSTEM 360 FORMATTED FILE SYSTEM (NIPS 360) FFS)

General Description

SUBMITTED BY:

CRAIG K. HILL
Captain, USA
CCTC Project Officer

APPROVED BY:

FREDERIC A. GRAF, JR.
Captain, U.S. Navy
Deputy Director,
NMCS ADP

Copies of this document may be obtained from the Defense Documentation Center, Cameron Station, Alexandria, Virginia 22314

ACKNOWLEDGMENT

This manual was prepared under the direction of the Chief for Programming with general technical support provided by the International Business Machines Corporation under contracts DCA 100-67-C-0062, DCA 100-69-C-0029, DCA 100-70-C-0031, DCA 100-70-C-0080, DCA 100-71-C-0047, and DCA 100-77-C-0065.

CONTENTS

ILLUSTRATIONS

ABSTRACT

This document presents an overall general description for the National Military Command System Information Processing System 360 Formatted File System (commonly referred to as NIPS or NIPS 360 FFS) an advanced data management system which is operational on IBM System/360 and System/370 computers. It describes the capabilities of the components which comprise the system: File Structure (FS), File Maintenance (FM), File Revision (FR), Retrieval and Sort Processor (RASP), Output Processor (OP), Quick Inquiry Processor (QUIP), Source Data Automation (SODA), Online Source Edit (EDIT), Terminal Processing (TP) and Utility Support (UT).

This document supersedes CSC GD 15-68.

Section 1

INTRODUCTION

The NIPS 360 PFS is an advanced data management system providing powerful, efficient, and flexible data management support to a wide variety of users. NIPS provides the ability to structure files, generate and maintain files, retrieve information and output that information in simple or complex arrays on a variety of output devices. Seven major processing functions are provided:

1. File Definition (structure)
2. File Generation and Maintenance
3. File Revision
4. Data Selection and Ordering (retrieval)
5. Formal Report Generation (output)
6. Terminal Processing and Display
7. Utility programs supporting table generation, subroutine loading and data conversion, among others.
8. Data Selection and Report Generation from non-NIPS data files.

NIPS functions under the Primary Control Program (PCP), the multiprogramming with fixed number of tasks (MFT) or the multiprogramming with variable number of tasks (MVT) versions of the IBM System/360 Operating System. NIPS also functions under IBM System/370 System Control Programming Operating System/Virtual Storage 1 (VS1) and Operating System/Virtual Storage 2 (VS2). Using multiprogramming capabilities, NIPS permits online functions to parallel the normal background or batched processing functions. NIPS can use a broad range of hardware since input/output device independence has been retained. Core sizes are relatively

noncritical in that NIPS will function within 90K of core memory. Core usage is dynamic, permitting NIPS to make maximum use of the partition or region of core provided by the operating system. Minimum core requirements for NIPS and the operating system are 128K for single partition systems and 256K for multitasking systems using the online capabilities. The design base was a S/360 Model 50H; however the instruction set utilization was limited to the standard instruction set and the decimal feature instructions.

NIPS will support data files organized with the Indexed Sequential Access Method (ISAM), resident on a direct access storage device (such as an IBM 3330, 2314 or 2311 Disk Storage Device), and the Sequential Access Method (SAM), resident on tape or direct access devices. When installed on S/370 with OS/VS, NIPS will support data files organized under the Virtual Storage Access Method (VSAM) for batch processing.

Section 2

FILE CONCEPTS

As in most generalized systems, WIPS is designed to permit nonprogrammers to specify data structures required to support a job function. This is done by a simple, meaningful language convenient to the user. The system performs exhaustive diagnostics and grants a high degree of independence from the physical storage of data, although users must be aware of the logical relationships between data elements. Terms used in this text are defined as follows:

Data base	-	A collection of data, supporting the general mission or function of the installation. It is typically composed of many different logical data sets (commonly called files).
Data file	-	A logical set of data elements, grouped into associated arrays called records.
Data file records	-	That collection of data elements identifiable by a unique data value or key.
Field	-	A single data element.
Secondary/keyword indexing	-	The capability to index, or access, the contents of a data file by a means other than the record identifier. The index data set is a logical data set of all index values and the data file records associated with each.

Data covering a broad spectrum of subjects may be received by an installation in varied forms and from many sources. This data is categorized by subject and added to a collection of similarly categorized data called a data file. To aid in placing each element of data into the file, and subsequently retrieving it, the files have a definite structure pattern called the file format. The file format

is designed to group similar data; i.e., description of an activity, event, person, or thing. Each of these groups is called a file record. Thus, a formatted file is an ordered collection of file records, each of which contains data arranged according to a previously established file format. In NIPS, the field is the smallest named entity processed. By use of a capability called partial-field notation, a user may refer to part of a field. However, direct reference in the simplest form defines field as the lowest-level element of a file record. The system provides for grouping contiguous fields in the record and defining a name for the resultant string. These composite, concatenated fields are called groups in NIPS terminology. Nesting of definitions (groups of groups of groups) is permitted and is effectively unlimited in depth of the resultant array. Data file records, therefore, may be defined as logical collections of fields and/or groups.

It is almost always necessary to carry both repetitive and nonrepetitive information in a file record; NIPS provides for this need (figure 1). Data that occurs only once per record is called fixed data. The data elements are termed fixed fields. A definable data pattern that recurs periodically within a record is called periodic data. Such fields are therefore termed periodic fields. Collections of field strings having the same format are called sets. Therefore, the string of fields containing nonrepetitive data in a file record may be defined as the fixed set. A maximum of 100 fields may be defined in a fixed set. Periodic fields, grouped and recurring as strings, are termed periodic sets, while one string of such a set is called a periodic subset. Up to 255 different periodic sets containing up to 100 fields each may be defined.

NIPS provides structures composed of a master segment or fixed set, with one level of subordination (the periodic set or sets). The format of a fixed set is constant throughout the file. More than one periodic set may be defined in the file structure, but as in the case of the fixed set, the format is constant throughout the file on a set-by-set basis. In other words, the format of the first periodic set in each record is the same, and the format of the second periodic set in each record is the same. This is valid for each record of the file. Formats of Periodic Set 1 and Periodic Set 2, however, need not be the same.

The system provides two other types of data storage. First, the user may include a variable field within the definition of each of his fixed or periodic sets. As the name implies, this field can contain variable length character strings and is included in physical storage only when there is actual data in the field. Second, the user may cause the system to structure several variable sets. Each variable set contains a variable length character string and exists only when data is present. The principal difference between the variable field and the variable set is in their association with other data elements. The data in a variable field is specifically associated with the data in a fixed set or a single periodic subset. Data in a variable set is a logical collection associated with, but subordinate to, the fixed set. The only relationship it has with periodic data is that of a common parent fixed set. Data may be added to or deleted from the variable set. Retrieval of records based on the content of variable data fields may be performed by either executing a scan of the field for the qualifying value(s) or by employing the keyword indexing capability. There is no limitation as to the length of a variable set.

Secondary and keyword indexing are optional capabilities available to enable a user to have more flexible control over his data. A file index serves essentially the same purpose as an index for a book. Given an abbreviated subject or topic, a book index lists all pages referencing that topic. Given a field value, a file index lists all data records containing that value. The purpose of such indexing is to reduce the time required to query a file. It does this by checking file index values specified by a query to determine which records must be examined in detail to satisfy the search parameters. Only these records rather than the entire data file are searched. Secondary indexing is the method for specifying fixed-length fields as indexes and retrieving records based on the contents of those fields. Keyword indexing is the method for specifying variable fields, variable sets and fixed-length fields as indexes; selecting values from the field as keywords, and; retrieving records based on the presence of the keywords within the queried fields.

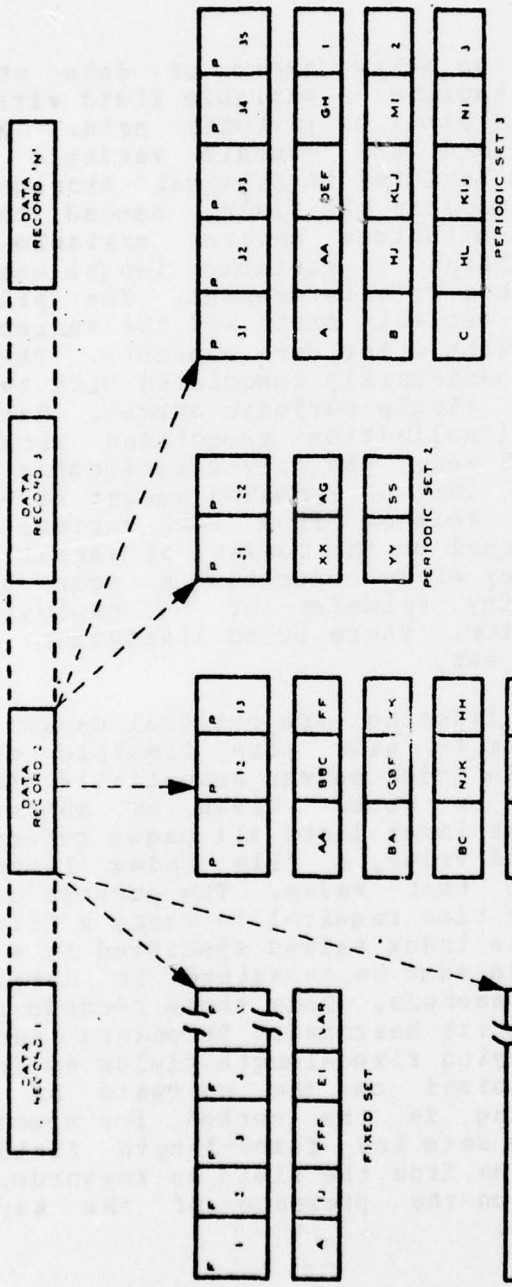

Figure 1. NIPS 360 FFS Data Record Organization

GENERAL DESCRIPTION

Data files which have been created by other data management systems or by special purpose programs can be processed by NIPS for data retrieval and report generation. This is a basic capability intended only as an initial aid in examining non-NIPS data files. To use the full capabilities of NIPS these data files should be converted to a standard NIPS data file.

Section 3

SYSTEM ORGANIZATION

The six major processing components and the general utility component each perform a general task and are associated with a general class of functions to be performed.

The File Structuring (FS), File Revision (FR), and File Maintenance (FM) components are primarily concerned with the creation, reorganization and maintenance of the system files. The Retrieval and Sort Processor (RASP) is the primary information retrieval tool while the Output Processor (OP) component provides for formal report production. Finally, the Terminal Processing (TP) component provides online NIPS programming, file interrogation, maintenance, and output capabilities through a variety of local and remote terminals. These include the IBM 3277, 2250, 2260 and 2265 display terminals, the IBM 1050 dial-up and IBM 2741 terminals. Each component, as well as the utility component, is discussed briefly in this section. The applicability of this system to a specific task area can only be evaluated at the detailed level. This description, therefore, suggests only in the most general terms the true depth of the actual software system.

3.1 File Structure

This system component permits the user to identify his data file and its security classification and to introduce definitions of the fields, groups, and sets that are to make up his data file. Although there is theoretically no limit to the number of unique data elements that can be named in a single file, the system imposes a practical limit of approximately 25,000 names. Structures are limited to a single level of subordination. At that level, however, up to 255 different periodic and/or variable sets may be specified.

There are also size restrictions in the length of a single periodic subset or fixed set. This constraint is 1,000 bytes in the current design.

File records (the collection of the fixed, periodic, and variable sets which are uniquely identified by a record identifier or key) are less easily defined as to size. The system performs dynamically, giving the user a high degree of flexibility. As a worst-case condition, the system may limit the user to 99,000 bytes per file record. However, the system loads into memory only those sets referenced by the job, thereby effectively permitting a significant expansion of record size.

During the file definition process, the user may desire to specify in advance certain automatic functions such as conversion of retrieval literals to coded file values, output data conversions or editing. If file indexing is desired, the user indicates which field or fields will provide the index values. The FS component provides mechanisms for this purpose and records these characteristics in the basic File Format Table (FFT). Record control or key fields may be defined as a single field or a string of multiple fields. The only limitation is that the total record control group may not exceed 244 characters. Periodic subsets may also have data fields assigned for sequencing and control within the record. If the user does not desire to provide these subset control elements, the system will create sequence numbers for the control function and will maintain them as a normal part of the FM function.

FS provides a simple, convenient way for the user to specify his file format and requirements. Using a simple free-format language which uses default options, the user can specify the logical data associations he desires.

Additional limitations to the FS component are as follows:

- A maximum of 50 subroutines/tables may be defined for a file.
- A maximum of 50 edit masks may be defined, each not more than 69 characters, and the

aggregate total of edit characters must not exceed 1,000.

- A total of 100 fields or groups may be defined for each set.
- A maximum of 99 fields may be defined in a group.
- A maximum of 50 groups may be defined in a set.
- Coordinate fields may consist of 5, 6, 7, 8, 11, or 15 characters only.
- General system conventions as to names and punctuation must be followed.

FS is also used to describe the format of any non-NIPS data file to be processed by NIPS. In describing the format of the file, the user must be aware of the various restrictions and limitations that affect non-NIPS processing. These include a maximum size of 1000 bytes in a single record format, the required presence of a non-repeating record format to serve as a fixed set, the presence of one or more fields in each record format to serve as record ID fields, the presence of a single field (up to 10 bytes) which uniquely identifies each record format. Also, the general restriction which apply to a standard NIPS file also apply to a non-NIPS file.

3.2 File Revision

The FR component provides the capability to revise the format in which data is stored in a NIPS data file. This is accomplished by the comparison of the FFT describing the file in its current format - to the FFT describing the file in its revised format. FR will generate, as a result of the comparison, FM logic statements which will be used by FM to copy selected data elements into the new format.

FR processes data on a field basis only. It will permit the addition, deletion, and relocation of fields as well as changes to their storage mode, size, and name. Periodic sets may be deleted or relocated. They may also be added,

but no data will be included, as a direct result of FR, in either new sets or fields. FR will permit fields from any set in the old file to be split into several sets in the new file. It will not permit fields from multiple sets in the old file to be merged into one set in the new file. There is no way for FR to logically connect the subsets of two existing sets. The FM logic statements generated may be modified by the user to suit his needs.

The only limitation is the need, on the part of the user, to recompile his FM logic statements and any retrievals, outputs, and terminal queries which might be affected as a result of file revision.

3.3 File Maintenance

This component provides the user with a broad and flexible capability for generating and maintaining data files. Using FM component features, the user may add, delete, or change file records, periodic sets or subsets, and variable sets. Also, the user may modify or change file fields and may change (increase or decrease) the volume of data associated with any file record. If indexing is specified for the file, the index data set is also maintained. This maintenance is automatic and transparent to the user. These functions may all be accomplished in parallel, or in a single pass of the master data file; they may be accomplished by virtually any mixture of unsequenced input transaction formats. Multiple file maintenance passes are permitted to support data dependent activities.

Input material may be records which are either fixed or variable in length and may be blocked or unblocked. Storage media for input material may be any supported readable media under the operating system.

All processing is controlled by logic statements provided by the user. These statements may be written in either a macro-like programmer-oriented language (POOL) or a high-level procedural English-like language (NFL) which is easy to learn and simple to use, yet powerful and flexible enough to efficiently accomplish a wide range of file maintenance functions. The user may perform functions using data from the input transaction, the master data file record, records from other NIPS files, or all three. The

GENERAL DESCRIPTION

Ordinary Maintenance (OM) feature of FM permits the user to write logic statements containing only OM transaction descriptor cards which can perform automatic validation of transaction data and unconditional updating of file fields. The OM statements can be combined with POOL statements to provide additional user flexibility. All input/output functions are performed automatically by the component; since the languages are indirect, the user makes his references using the field names he has assigned as meaningful titles of data elements. The languages include instructions that permit automatic table validation or conversion or automatic linkage to user-provided, special-purpose subroutines. The indirect-addressing capability, used for processing field-initiated corrections, is another FM feature.

In addition to maintenance (or generation) of a data file, this component permits the user to create auxiliary output files simultaneously with the maintenance process. The system will permit the generation of two independent print streams, two independent punch streams, and five auxiliary sequentially-organized output streams (tape or direct access volumes) in parallel with the prime processing function.

Generative code techniques are used in the actual data manipulation. To avoid wasteful regeneration, the system provides a complete, automatic library maintenance function for the user's logic statements. Maintenance execution, therefore, may consist of executing prestored logic statements, compiling logic statements and library updating, or combinations of these functions. There is no limit to the number of different statements that can be precompiled and retained in the library, or to the number of statements that can be used in a single file update run.

A data file may be maintained as an ISAM, SAM or VSAM data set. Large SAM files may be segmented into volumes of data records containing a range of record keys. These may then be updated and queried individually or in groups. For a SAM file that is indexed, volume control information is provided so that queries against the file need pass only those volumes containing the candidate records appropriate to the query instead of the entire data file.

Other component functions provide for the creation of summaries, run logs and/or audit trails during file processing. These functions are specified by the user and are not limited to a set of predefined system capabilities. They are as simple or complex as the user's requirements dictate.

Limitations to FM include the following:

- Maximum size of transaction record is 1,000 characters.
- A maximum of 24 logic statements may be compiled in any one execution of FM.

3.4 Retrieval and Sort Processor

The RASP component, primary retriever of the system, is characterized by its voluminous, batched-processing capabilities. Using generative code techniques, RASP includes an automatic library maintenance capability which permits broad utilization of the standing query. Compile-and-go modes of operation permit satisfaction of the ad hoc query requirement and complete intermingling of compile-and-go and standing query processing in a single job process is permitted.

RASP uses a simple English-like, free-format condition/action language which is flexible in notation. By specifying the retrieval condition, the user may select specific records to be retrieved from the data file. Comparison operators provided are the normal equal to, less than, greater than and between. All may be preceded by the negative operator "not." Boolean connectors are permitted, as are up to eight levels of nested parentheses. Two geographic retrieval operators are also provided: irregular area and circle search. The former is of interest in that it provides area-to-area, area-to-line, line-to-line, and point-to-area capabilities and is not constrained to convex area definitions.

The true power of the processor, however, is not directly indicated by the language. The scope of the processor is more directly related to its ability to provide many answer sets, each of different control or sequence, by

a single pass of either the entire data file or only those records selected as possible candidates for retrieval by index processing. In RASP, the simplest qualification statement is called a condition. Satisfaction of the condition assigns the output of the file record (or a selected portion of the file record) to a temporary file for later system sequencing. Since the user must frequently output information from a file record in several sequences, each condition statement may have several associated sort statements. Each sort statement is simply the string of fields and/or literals upon which the user desires to sequence the answer records. Conditions may be prestored with their associated sort statements on the query library. Each entry on the library is termed a query title. Since the user may need to define several different criteria in selecting qualifying records, the capability to associate different conditions with a single title is desirable. Thus the system permits multiple conditions per title entry. The total number of conditions and sort designations may be as large as 255 per title entry. Finally, in a single file pass, the system will accept up to 96 title requests. In short, the user could process 96 times 255 answer sets in a single file pass. This limitation is consistent with the overall system philosophy to establish limitations on a volume or size level that the user normally will never reach. Index processing will be used by RASP when specified for the file and desired by the user at run time. RASP will analyze queries for conditions against index fields, will then examine the index data set, and obtain a list of records which contain the appropriate values as specified in the queries. Only these records, rather than the entire data file, will be examined in detail.

Most retrievals compare field contents to literals provided in the query statement. The system, however, permits restricted comparison of field contents, one field to another, as well as the normal literal mode. Automatic conversion of comparison literals from external form to file form is provided, and partial field notation and/or literal masking is also included.

From a programming viewpoint, the result of the RASP process is slightly different from that sensed by the using analyst. Although RASP gives the result that would be achieved by redundantly outputting data records after each

qualification, in practice this redundant output is avoided. Instead, a technique is utilized that permits sequencing of the resultant answers by pointers rather than physically sorting the data records themselves. Sequencing time improvements of well over 10 to 1 above normal file sort times can thus be achieved. In applications where a retrieval results in high volumes of output, this attribute is critical. The result of RASP processing is a subset of the interrogated file; after processing by one of the MIPS utility programs, this subset may be further maintained, retrieved from, and used as input to update another data file.

Additional features of the component are:

The ability to modify retrieval values and sort field specifications at execution time through use of the skeleton query feature

The use of user-written subroutines to qualify data through use of the FUNCTION operator

The passing of information generated by user-written subroutines to the output processors through use of system-provided work areas.

Limitations imposed on RASP as are follows:

- Maximum number of SORT or SELECT pairs per IF or FURTHER statement is 20.
- Maximum number of RIT names per retrieval is 50.
- Maximum number of statements per retrieval is 256.
- Maximum number of subroutines per retrieval is 20.
- Maximum number of clauses per statement is 256.
- Maximum number of retrievals in a single file job is 96.

- Maximum number of retrievals in a merged file job is 1.
- Maximum number of files in a merged file job is 10.
- Maximum number of field or group names per retrieval is 400.
- Maximum number of replaceable variables per retrieval is 255.
- Maximum number of cards per statement is 27.
- Maximum number of characters constituting a list of multiple data values in a clause is 250 total characters in the list plus 3 for each value.
- Maximum number of operands for all SORT statements in one RASP job is 340.

3.5 Output Processor

This large component is a report generator. As such, it includes capabilities for formatting extremely complex and lengthy reports. In addition to the normal page formatting capabilities, the OP includes a complete logical conditioning capability, computational capabilities, summarizations, totaling, and subtotaling. In addition to the computational capabilities, one and two dimensional sum and count matrixes may be specified. It is a complete and comprehensive report generation system driven by a master data file or by the result of the retrieval process. An interfile output capability allows access to secondary file records during the output process. Data elements from these secondary files may be combined with primary file data to produce a composite report. Records are selected from secondary files by means of pointers within the primary file. Output may be in print, on card, disk, or tape, or it may be any combination of these. Classification cover pages, downgrading information, and warning indications in the case of classification differences are automatically provided by the component.

GENERAL DESCRIPTION

A flexible, free form user language is provided. As in the other components, generative code techniques are utilized to produce a Report Instruction Table (RIT). Therefore, OP includes a complete, automatic library capability. Debug modes assist in the checkout of complex report formats. Implicit and explicit data conversion and/or editing is also included. Any number of reports may be published from a file or RASP output in a single run.

Maximum limitations to the OP component are as follows:

- 40 reports may be structured in one OP run.
- 10 data files may be referenced in one report.
- 42 sets may be referenced in one report.
- 30 subroutines may be specified in one report.
- 550 named literals may be defined within one report.
- 52 characters is the maximum literal size.
- 8,192 bytes constitute a compiled individual level 1 statement.
- 5 retriever-generated work area values are used per set; each one is a full word in length.

3.6 Terminal Processing

The TP component provides NIPS users with an efficient, flexible means of operating in a terminal environment. This component provides the NIPS interface to support local IBM 3270 Information Display Systems, local or remote IBM 2260 Display Stations, local IBM 2250 Display Units, the IBM 2741 Communication Terminals, and the IBM 1050 Data Communication System (Dial-up). The number of devices supported is limited only by the hardware configuration and the loading of the system. The TP component permits utilization of these devices in a true online mode, allowing different tasks to be initiated from the devices without scheduling

the tasks through the normal operating system job stream. The component consists of two major functional sections, the TP Monitor and the TP Supervisor.

The TP Monitor is normally core resident in a single partition or region. This section of the component services device requests, performs input/output queueing, and controls task assignments. Depending on user environmental requirements, one or more TP Supervisor(s) may reside in machine core. Their primary function is simply to provide interface for TP application programs. Output generated by the application programs is made available to the terminal user for paging in the conversational mode.

NIPS provides the following terminal application programs or components:

a. Quick Inquiry Processor (QUIP) is designed to give the user a powerful data retrieval and display capability.

b. Source Data Automation (SODA) provides the user with a means of executing a precompiled file maintenance logic statement against an update transaction entered by an operator at an online terminal.

c. Online Source Edit (EDIT) provides the ability to enter, manage and edit NIPS and ALC source code language statements at the terminal and to submit jobs to the operating system for batch processing.

d. Utility programs provide the ability to send messages and data between terminals and to communicate with the computer operator.

e. FORMATTER allows the user to define a display format for a CRT and to use this format at the terminal to enter data. This data can then be used by the NIPS application programs FMSODA, QUIP, BLAST, EDIT, or a user written program.

The QUIP language consists of a simplified output language and a subset of the RASP language. Although QUIP was designed primarily for terminal application, it can also

be operated in the batch mode to provide a simple, efficient tool to process ad hoc queries. When executing QUIP as a problem program in the online terminal environment, it requires a direct access resident ISAM NIPS data file or a direct access resident SAM data file. In the online environment, QUIP can also create and subsequently interrogate a subfile of a NIPS file. A NIPS data file organized as either ISAM, SAM or VSAM organization (tape or disk resident) is acceptable when QUIP is executed in the background environment. Where possible, QUIP will attempt to limit the number of data records that must be examined in detail through the use of file indexing in the same manner as RASP. The QUIP language allows user or system formatted output lines. One and two dimensional sum and count matrixes can easily be specified in addition to computational operations. An additional capability allows the terminal user to select an OP RIT from the user library and display pages of the output report on the terminal. QUIP provides an interfile output capability in the same manner as OP. Data elements extracted from secondary files may be combined with primary file data to produce a composite report. QUIP can also be used to retrieve and output data from a non-NIPS data file. Non-NIPS processing can be performed in the online or batch environment against either SAM, ISAM or VSAM files. Many of the QUIP capabilities available for processing a NIPS file can also be used for processing a non-NIPS file, including interfile output processing.

The SODA component uses a free-format keyword language consisting of control statements which identify files, records, report formats, corrections, and disposition actions. Given a file, SODA will fetch the record identified in the update transaction, and provide both the record and the transaction to the library-stored logic statement which performs the required edit and update function. The record is returned to the data file in updated form under operator control. The user's input transaction is validated by the logic statement during execution. If errors are detected, immediate notification may be given to the terminal operator. The user may correct individual lines of his transaction, using the standard TP correction procedures and repeat the SODA request. The user may alternatively enter key-numbered corrections to individual fields and repeat the request. SODA will

maintain file indexes, if present, in the same manner as FM. SODA cannot be used to perform file maintenance on a VSAM data base.

The EDIT component provides the capability to enter and maintain source language statements from the terminal. The source statements may be stored on a library and modified from the terminal. The modified source statements may then be edited and any errors reported back to the terminal operator. When errors have been corrected, a Job Control Language (JCL) stream may be constructed and submitted to the operating system as a background job.

The NIPS TP utlity programs include the capabilities to dump output to specified printers, route output to other terminals, and send messages to other terminals or to the computer operator. The online capability is also provided to access a distribution data set for immediate review of outputs previously stored as the result of batch or online report generation.

The FORMATTER component provides the user capability to specify the attributes of the CRT display image using the FORMATTER language. The specification of the display image includes the definitions of data to be displayed on the CRT, the definition of data to be placed on the IMQ (Input Message Queue) records, and the action to be taken after the CRT operator enters data on this display. The user display format can be defined and stored in either the batch or online mode. The stored format is utilized in the operational phase. The CRT operator can invoke FORMATTER and then call a format by the user assigned name. FORMATTER will write the skeleton image on the CRT and wait for the CRT operator to key data into the defined fields. When the operator has finished, FORMATTER will create the IMQ records according to the user specifications. FORMATTER will then invoke a NIPS application program or it will generate a new format display, depending on the specifications in the format definition. The capability to specify subsequent formats and application programs, combined with a capability to pass data from QUIP and SODA to a new format, provides extensive flexibility and power to control the chaining of online applications.

3.7 Utility Components

A large collection of utility support programs is provided by NIPS. A data validation/conversion-table generation capability which assures efficient use of this important data processing tool is provided. Tables formed by the system are modular so that usage on minimum core environments may be achieved. Table sizes may effectively be considered up to approximately 528,000 bytes of argument/function pairs. They provide simple, effective utilization of validation techniques not otherwise possible. Maximum combined size of the argument and function is 256 bytes.

A utility is provided to allow the placing of load modules, created by compiling user-written subroutines, on a NIPS library. The subroutines will contain the linkage necessary for proper utilization by NIPS. The subroutines may be written in Basic Assembler Language, COBOL, FORTRAN, PL/1, or JOVIAL.

Utilities are provided to convert the modes of NIPS files from ISAM or VSAM to SAM and vice-versa. Other utilities are available to specify and create indexes for existing files, and to transfer index data sets from disk to tape and vice-versa. Utilities are also available to analyze text for keyword selection and to create and maintain keyword dictionaries and tables. Another accepts a retrieval answer file and converts it to a data file containing all of the properties of a NIPS master file.

Section 4

SYSTEM FEATURES

To simplify the training of nonprogramming oriented users, the system makes extensive use of the stored procedure and cataloging capability of the operating system. This results in effectively eliminating the requirement for the typical user to code JCL streams to control his job. The skilled programmer may, however, continue to use fully the flexibilities provided by the operating system.

Modularity of programming ensures simplified maintenance and error correction in the operating software. Relatively unique code generation techniques ensure ease of modification in the user language areas, if needed.

All user-generated material required to support a data file is contained as part of the data file or user library. The FFT, FM logic statements, and data records are all a part of the data file itself. All conversion tables, subroutines, RASP queries, QUIP queries, and OP RITs can be on a single user library. The user's source material may reside in the same library or in a separate library. By causing the library to occupy a direct access device with the data set, a notable degree of file transportability is achieved. A system user may effectively move a single disk pack (data volume permitting) to another system operating with NIPS and be assured that he has all the materials necessary to maintain, retrieve, or display data from his file.

File analysis and run optimization statistics may be produced by FM, RASP, QUIP and OP upon demand. File analysis statistics assist a user in redesigning a file and selecting proper data elements for secondary indexing. Run optimization statistics assist in tuning production jobs to run most efficiently.

Eleven volumes of user-oriented documentation cover the system components, error codes, procedures, techniques and general file concepts. Program Specification Manuals are

available to assist installation system programmers in understanding system capabilities.

Section 5

OPERATING SYSTEM CONSIDERATIONS

NIPS will operate under five operating system configurations:

PCP	Primary Control Program. The standard operating system which processes one job at a time in sequence.
MFT	Multiprogramming with a fixed number of tasks. A partitioned core system allowing up to four jobs to operate concurrently in independent fixed-size partitions.
MVT	Multiprogramming with a variable number of tasks. A regional core system allowing up to 15 jobs to operate concurrently in variable-size regions.
VS1	Operating System/Virtual Storage 1. A virtual storage system which can perform 15 different tasks concurrently.
VS2	Operating System/Virtual Storage 2. A virtual storage system which can perform 63 different tasks concurrently.

NIPS is not restricted to any one level of the operating system and it will use subsequent releases as they become available. Multiprogramming capabilities of the operating system permit multiple jobs to operate concurrently in independent partitions, regions or virtual storage. The data management facilities of the operating system will handle all system and user data storage. NIPS data files can be organized under ISAM for direct access device files, or SAM for tape or direct access files, or under VSAM for direct access files on the S/370 operating under VS1 or VS2. Data file indexes, where applicable, are organized under the Basic Direct Access Method (BDAM) and reside on a direct

access device. The various system libraries will be stored using Basic Partitioned Access Method (BPAM). The Basic Graphic Programming Services of the operating system are used for the TP component that services the local IBM 2250 and 2260 display terminals. The Basic Telecommunications Access Method (BTAM) is required for remote device support and for local IBM 3270 Information Display Systems.

In addition to the basic operating system, the following must be included at system generation time:

a. Programs

1. Assembler - F level
2. Linkage Editor
3. Sort

b. Libraries

1. MACLIB - Macro Library
2. SORTLIB - Sort Library

c. Data Management Access Methods

1. Sequential Access Method
2. Index Sequential Access Method
3. Basic Direct Access Method
4. Basic Partitioned Access Method
5. Basic Graphics Programming Services (needed only for terminal processing support)
6. Basic Telecommunication Access Method (needed only for remote terminal processing support)
7. Virtual Storage Access Method (available only on an S/370 with VS1 or VS2)

Section 6

HARDWARE CONSIDERATIONS

NIPS has been designed and programmed for an IBM System 360 Model 50H (256K core size). It will also operate on a Model 40H and larger models of the System 360 line and Model 135 and larger models of the System 370 line. Without online terminals, it will operate on a 128K core size machine, however, experience has shown that the H core is the minimum for complex applications on nonvirtual storage systems. NIPS can use magnetic tapes, direct access devices, card reader/punches, online printers, a console typewriter, and IBM 2260, 2265, 2250, 3277, 3275, 1050, and 2741 terminals

A minimum configuration could include three IBM 2311 disk units, (although an IBM 2314 or 3330 disk storage unit is considered highly desirable) a card reader, and an on-line printer. Tape requirements are related to the user's requirements and range from none to a quantity sufficient to perform the largest sort the user may require. Disk sorting is used to the capacity of the direct access devices available to the system.

The system will service a number of IBM 2848 display control units, each of which will support one 1053 printer and eight local 2260 terminals for online processing. Up to 32 IBM 3277 display stations may be attached to the system via each 3275, and 3272 control unit. An unlimited number of local 2250 terminals are supported under these configurations:

a. 2250 -1 Each terminal has its own control unit.

b. 2250 -3 Up to four terminals for each 2840-2 control unit.

Remote IBM 3277, 3275, 2260, 2265, 1050, or 2741 configurations must include appropriate communications facilities and 27ØX data adapter or transmission control units.

Section 7

DEVELOPMENT AND MAINTENANCE

The NIPS data management system is managed by the Defense Communications Agency to meet the requirements of the National Military Command System. In the interests of economy and standardization NIPS is provided on a fully supported basis to other Department of Defense organizations. Development and maintenance activities are directed in consideration of the needs of this fully supported NIPS user community.

The requirements of the command and control and intelligence communities call for a very high level of reliability. The goal of NIPS maintenance is to insure efficient performance to meet all capabilities specified in the user manual series. Designated fully supported users are encouraged to submit fully documented Discrepancy/Change Reports (DCR) for discrepancies between system operation and documented performance specifications. Each discrepancy is acknowledged with corrective action for programs, documentation or user application. All NIPS users enjoy the benefits of a maintenance program based on a wide variety of applications.

New developments and the results of system maintenance are provided to NIPS users in the form of tested system releases. Releases are scheduled depending on the magnitude of the changes included in the release, major Department of Defense activities and avoidance of concurrent implementation with a new operating system release. Emergency fixes are provided immediately, if possible.

An aggressive enhancement program has been underway for several years and is planned to continue. Major areas toward which development activities have been directed are as follows:

a. Efficiency. Continuing efforts will be directed toward optimizing use of computer resources for the multiprogramming environment.

b. Interfile operations. Multiple file reference both during update and output will support applications integration.

c. Transaction Management. Improved capabilities to route and edit transactions entering and leaving an installation will reduce manual transaction handling and point toward real-time information handling.

d. Language Standardization. Consistent use of punctuation and instruction forms will reduce user errors and speed training.

e. Extended OnLine Operations. Continued expansion of remote terminal services will place the ultimate information user in closer contact with his information.

f. Statistics. Extensive information will be available about the operation of applications to enable users to tune applications for greater efficiency.

g. Capability Improvement. A continuing effort is underway to make the system easier to use and extend the range of capabilities available to the user.

h. Documentation. Complete and easily used documentation is the key to effective use of NIPS. Constant attention is given to current maintenance and improvement to users manuals and other system documentation.

The enhancement program is managed so as to provide constantly improving capability with minimum disruption to the operation of current user applications. The following are major criteria applied to the design and implementation of enhancements.

a. Incorporate capabilities so that each module will perform its basic operation in a 90k byte partition or region. New capability which if used causes an

application to exceed this limitation must be optional.

b. Maintain dynamic core utilization capabilities so that NIPS applications will efficiently use additional core, if available.

c. Maintain the transportability and transferability of NIPS applications between NIPS using installations.

d. Maintain standard operating system conventions.

e. Maintain an efficiency balance between update and retrieval.

f. Minimize the degree to which an installation can become I/O or compute bound.

g. Incorporate only those capabilities that are documented for easy understanding by the average user and foolproof in operation.

h. Provide a full range of diagnostic, debugging, and tuning tools to the user.

i. Maintain a high degree of generalization while incorporating optional specialized capabilities which optimize the use of NIPS for Department of Defense command and control and intelligence functions.

j. Provide for continued maintainability so as to insure a very high level of reliability during the development and operation of applications.

k. Incorporate new sophisticated capabilities so that:

- Changes to existing applications are minimized.
- Simple basic NIPS capabilities are not lost.
- Standard default options are available.

Section 8

SUMMARY

NIPS exemplifies the heavy-duty file processor which has been the mainstay of the Department of Defense command and control and intelligence data handling. In its current version, this system incorporates a comprehensive online capability which further enhances its adaptability to today's processing requirements. The evolutionary approach continues to be the foundation of the system's development. The validity of this approach is increasingly apparent as the number of users and applications increases. For application areas with high-volume and large file-processing requirements, NIPS provides a convenient, efficient, and flexible method of solving the data handling problems in the IBM System 360 and System 370 hardware and software environment.

DISTRIBUTION

CCTC CODES	COPIES
C124 (Reference)	2
C124 (Record Copy)	1
C240 (COR for CSC)	20
C300	1
C310	1
C311	1
C312	1
C313	1
C314	1
C315	1
C316	1
C317	1
C320 (Training)	5
C321	1
C322	1
C323	1
C324	1
C325	1
C340	5
C341 (Maintenance Contractor)	8
C341 (Stock)	50
C700	1
C702	1
C703	1
C705	1
C710 (Tech Ser Support)	2
C720	1
C730	2

DCA CODES	
C100	1
C110	1
C205	1
C530	1
C600	1

EXTERNAL	COPIES
Director of Administrative Services, Office of the Joint Chiefs of Staff Attn: Chief, Personnel Division, Room 2A944, The Pentagon Washington, D.C. 20301	1
Director for Personnel, J-1, Office of the Joint Chiefs of Staff, Attn: Chief, Data Service Office, Room 1B738C, The Pentagon, Washington, D.C. 20301	1
Director for Operations, J-3, Office of the Joint Chiefs Staff, Attn: Chief, Data Processing Division, Room 2C869, The Pentagon, Washington, D.C. 20301	1
Director for Operations, J-3, Office of the Joint Chiefs of Staff, Attn: P & AD, Room 2B870, The Pentagon, Washington, D.C. 20301	1
Director for Operations, J-3, Office of the Joint Chiefs of Staff, Attn: Deputy Director for Operations (Reconnaisance and Electronic Warfare) Room 2D921, The Pentagon, Washington, D.C. 20301	1
Director for Logistics, J-4, Office of the Joint Chiefs of Staff, Room 2E828, The Pentagon, Washington, D.C. 20301	1
Chief, Studies Analysis and Gaming Agency, Attn: Chief, Force Analysis Branch, Room 1D928A, The Pentagon, Washington, D.C. 20301	1
Automatic Data Processing, Liaison Office, National Military Command Center, Room 2D901A, The Pentagon, Washington, D.C. 20301	1
Automatic Data Processing Division Supreme Headquarters Allied Powers, Europe Attn: SA & P Branch, APO New York 09055	1
Defense Civil Preparedness Agency, Computer Systems Division, Room 3D317, The Pentagon, Washington, D.C. 20301	1
Director, Defense Communications Agency, Office of MEECN System Engineering, Attn: Code 960T, Washington, D.C. 20301	1
Director, Defense Communications Engineering Center, Hybrid Simulation Facility, 1860 Wiehle Avenue, Reston, VA 22070	1

EXTERNAL	COPIES
Commander, Joint Technical Support Activity Attn: Chief, Software Operations Division 1860 Wiehle Avenue, Reston, VA 22070	1
Director, Defense Intelligence Agency Attn: DS - 5C2 Washington, DC 20301	5
Commander-in-Chief, Pacific, Attn: J6331, FPO San Francisco, 96610	1
Commander-in-Chief, Europe, Attn: Chief, EUCOM ADP Systems Office (ECADP) APO New York 09128	1
Commander-in-Chief, US Army Europe and Seventh Army Attn: ODCS, OPS APO New York 09403	1
Commanding General, US Army Forces Command, Attn: Data Support Division, Building 206, Fort McPherson, GA 30303	1
Commander, Fleet Intelligence Center, Europe, Box 18, Naval Air Station, Jacksonville, FL 32212	1
Commanding Officer, Naval Air Engineering Center, Ground Support Equipment Department, SE 314, Building 76-1, Philadelphia, PA 19112	1
Commanding Officer, Naval Security Group Command, 3801 Nebraska Avenue, NW, Attn: GP22, Washington, DC 20390	1
Commanding Officer, Navy Ships Parts Control Center Attn: Code 712, Mechanicsburg, PA 17055	1
Headquarters, US Marine Corps, Attn: System Design and Programming Section (MC-JSMD-7) Washington, DC 20380	1
Commanding Officer, US Army Forces Command Intelligence Center, Attn: AFIC-PD, Fort Bragg, NC 28307	1
Commander, US Army Foreign Science and Technology Center Attn: AMXSJ-CS, 220 Seventh Street, NE, Charlottsville, VA 22212	1

EXTERNAL	COPIES
Commanding Officer, US Army Security Agency, Command Data Systems Activity (CDSA) Arlington Hall Station Arlington, VA 22212	1
Commanding Officer, US Army Security Agency Field Station - Augsburg, Attn: IAEADP, APO New York 09458	1
Commander, Fleet Intelligence Center, Atlantic, Attn: DPS, Norfolk, VA 23511	1
Commander, Fleet Intelligence Center, Pacific, Box 127 FPO San Francisco, 96610	1
Air Force Operations Center, Attn: Systems Division (XOOCSC) Washington, DC 20301	1
Commander, Armed Forces Air Intelligence Training Center, Attn: TINI-MIT, Lowry AFB, CO 80230	1
Commander, Air Force Data Services Center, Attn: Director of System Support, Washington, DC 20330	1
Commander-in-Chief, US Air Forces in Europe, ATTN: ACDI APO New York 09332	1
Commander, USAF Tactical Air Command, Langley AFB, VA 23665	1
Commander, Space and Missile Test Center, Attn: (ROCA) Building 7000, Vandenberg, AFB, CA 93437	1
Naval Air Systems Command, Naval Air Station, Code 13000, Jacksonville, FL 32212	1
Commanding General, US Army Computer Systems Command, Attn: Support Operations Directorate, Fort Belvoir, VA	1
Defense Documentation Center, Cameron Station, Alexandria, VA 22314	12
TOTAL	171

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: General Description
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER

(6) 4. TITLE (and Subtitle): NMCS Information Processing System 360 Formatted File System (NIPS 360 FFS) - General Description

5. TYPE OF REPORT & PERIOD COVERED

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

8. CONTRACT OR GRANT NUMBER(s): DCA 100-77-C-0065

9. PERFORMING ORGANIZATION NAME AND ADDRESS: International Business Machines, Corp. Rosslyn, Virginia

10. PROGRAM ELEMENT, PROJECT, AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS: National Military Command System Support Center, The Pentagon, Washington, D.C. 20301

(11) 12. REPORT DATE: 1 September 1978

13. NUMBER OF PAGES: 40

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Copies of this document may be obtained from the Defense Documentation Center, Cameron Station, Alexandria, Virginia 22314.
This document has been approved for public release and sale; its distribution is unlimited.

(12) 47 p.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

(14) CCTC-CSM-GD-15-78

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

(9) Computer systems manual.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
This document presents an overall general description for the National Military Command System Information Processing System 360 Formatted File System (commonly referred to as NIPS or NIPS 360 FFS) an advanced data management system which is operational on IBM System/360 and System/370 computers. It describes the capabilities of the components which comprise the system: File Structure (FS), File Maintenance, File Revision, Retrieval and Sort Processor, Output Processor, Quick Inquiry Processor, Source Data Automation, Online Source Edit, Terminal Processing and Utility Support.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

409 658